

Multi-Objective Optimisation on Transportation Networks

Dan Costelloe, Peter Mooney and Adam Winstanley.

Department of Computer Science,
National University of Ireland Maynooth (NUIM),
Co. Kildare. Ireland.

Telephone: +365 1 123456 Email:
{danc, pmooney, adamw}@cs.may.ie

Abstract

A multi-objective optimization scheme for transportation planning is described. Multi-criteria analysis provides a major advantage in its ability to take into account a range of different, often unrelated criteria, even if these criteria cannot be directly related to quantitative outcome measures. The approach described is specifically addressed to public transportation networks but it is also applicable to other types of physical networks including computer networks. Research into developing an evolutionary approach to public transportation network optimisation, by use of a carefully chosen fitness function, is outlined.

1. Introduction

Transportation analysis within a GIS (Geographic Information System) environment has become common practice in many application areas. Transport, by its very nature, lends itself to a multi-disciplinary study. An every increasing need for complex path algorithms and path computation has developed from the rapid emergence of GIS systems such as intelligent vehicle systems. Optimal route planning is made complicated by the existence of factors such as multiple modes, planned arrival and departure schedules, multiple fare structures, dynamic changes to the network. Route or journey planning is the systematic search through a transportation network to find an *optimal* journey specification. This specification is nearly always required to satisfy some initial set of constraints (times, road types, costs). The set of constraints is better described as a preference for particular routes and departure/arrival times and desired departure and destination locations. Constraints may be placed on variables or criterion that are easy to quantify, for example departure and destination time. However, other criteria are more difficult to quantify. Examples include preference for certain modes or transport and preference for particular road types.

Optimal journey specifications can now be defined as a journey specification exhibiting minimal values for all variables (criteria, objectives) considered. However, humans are seldom capable of discovering these optimal solutions unless the network search space is relatively small. It is very often the case that comprehensive searches are too expensive in terms of information gathering and retrieval and search time. To avoid the costs involved in the searching process, in terms of effort and time, humans will only attempt to find *any* satisfactory journey specification. Behavioral scientists define the term *satisficing* (Nijkamp and Van Deft 1971) for this type of human behaviour in regard to information searching and decision-making in large search spaces. Humans are risk averse in selecting alternative journeys when a journey specification that satisfies certain minimal, weak, conditions and criteria has been found.

We propose a series of computer-aided techniques to assist travellers in searching for and planning more efficient journeys on a transportation network. A given (transportation route finding) problem may have a set of solutions, some good, some not so good. Within this set of solutions (if it exists), there also exists a subset of optimal (best) solutions. Depending on the problem instance, there may be one optimal solution or a group of them. Using these techniques we can find the best solution or set of best solutions corresponding to journey constraints and optimisation criteria. Optimisation may now be redefined as the task of finding the(se) best solution(s).

2. Criteria and Objectives

Early routing models typically used the standard linear programming techniques to optimise an objective function consisting of a single criterion or a weighted combination of multiple criteria. This type of approach to multi-objective routing does not allow a complete analysis of trade-offs between the various criteria when some weighted combination is optimised. The weighted combination approach does not in any way guarantee that all non-dominated paths will be discovered. It is those non-dominated paths that describe journey specifications exhibiting minimal values for criteria such that they cannot be bettered by other journey specifications on all criteria. A simple example provides motivation for these claims. Suppose that a journey from A to B is described by a 3-D vector AB with elements [time required, financial cost, distance] and that for a particular network model (with A and B defined) this vector is [100, 10, 200]. Another vector AB* (also describing a journey from A to B) has elements [120, 5, 210]. If one compares the vectors AB and AB* on a weighted combination or sum of elements it is easy to determine the *minimal* journey specification. However, this naïve analysis yields a decision on optimality that is not taken with respect to the entire decision variable space. As the vector dimensions grow larger to incorporate more criteria and objectives this analysis becomes more unpredictable and unstable.

When more than one optimisation criteria are involved, aggregate-sum approaches are often applied to condense the multiple objectives into one to make the optimisation process easier. For some problems, however, this approach may not be feasible, as trade-offs may exist between criteria: an increase in one may result in a decrease in another (some or all of the time), depending on the values of the other criteria. It is in cases like this that a technique called Multi-objective Optimisation may be employed. For a survey of Evolutionary Multi-objective Optimisation techniques, see (Coello 1999). Optimisation techniques are used to achieve the 'best' (or as close to the 'best' as possible) solution(s) to a given problem. The 'best' solution may be the one that takes the least amount of time to compute, costs the least financially or achieves the highest score according to some evaluation scheme. Many optimisation techniques involve navigating the *search space* for an optimal solution. Some problems have very large search spaces, meaning that simple, brute-force searches are too complex in terms of the time it would take (or the necessary resources) for the search to complete. It is for this reason that approaches such as Tabu Search (Glover 1990), Simulated Annealing (Metropolis and Rosenbluth 1958), and Genetic Algorithms (Holland 1975) (among others) have evolved.

3. Multi-Objective Optimisation.

In multi-objective optimisation problems, each objective may be represented as a vector entry, with the vector itself representing a solution, for example:

$$X \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

represents a single solution to an n-objective optimisation problem. Given two vectors (X and Y), we say that $X <_p Y$ (X is partially less than Y or X dominates Y) if:

$$X <_p Y \Leftrightarrow \forall i(x_i \leq y_i) \wedge \exists i(x_i < y_i)$$

Solutions which are non-dominated (the *Pareto-optimal* set) can be considered as better overall solutions than those in the whole set as they have no other solutions that are better than them on all criteria. While this process doesn't necessarily identify any one outstanding solution, it does narrow down the search space to a set of solutions, which may be more easily navigated.

Multi-modal, multi-criteria optimised route planning is not yet well studied in the literature. Methods for solving single objective planning problems have been studied extensively for the past 40 years. However, almost every important real world problem involves more than one objective. Multiple objective optimisation problems are similar to single optimisation problems except that they have a stack (vector) of objectives (criteria) to optimise rather than just a single one. (Costelloe et al. 2000) provides a solution methodology for multi-objective optimisation of routes on a static network model with at least three objectives.

3.1 Producing the Pareto-optimal set

Our implementation produces a set of candidate solutions C from a graph G (model of the public transportation network) and extracts the Pareto-optimal set P from C . Each member of C has an associated path description vector of the form:

$$C_{ab} \begin{bmatrix} time \\ cost \\ changes \end{bmatrix}$$

Each candidate represents a journey from a to b with its associated time, cost and number of modal changes. Of course other choices of criteria are allowed depending on the situation. P is then obtained by searching the set C for non-dominated solutions. The solutions in P are considered better than those that are not in P , because their path description vectors cannot be bettered on all criteria. Once P has been constructed, the decision-maker must then choose which P_{ij} to use. With this approach the path description vector may be extended to handle more entries (objectives) to analyse tradeoffs between these entries.

3.2 Computing the Pareto Optimal Set.

The first step in computing the Pareto Optimal Set is to produce the set of candidate solutions C . Each node in the graph model $G = (V,E,S)$ is either an intermediate stopping point or major station (intersection) on the public transport network. The set S represents the set of

routes operating on the entire network. Nodes can then be partitioned into departure destination pairs. Then for every departure destination pair, several different paths must be computed. These are deemed the candidate solutions for the optimal path(s) between the departure and destination nodes. Individual candidate solutions are obtained by running several different route finding algorithms each with different strategies for finding the shortest, cheapest etc path between the pair of nodes in question. Examples of the route finding algorithms implemented are A*, Dijkstra's Algorithm and Bellman-Ford-Moore. Each of the algorithms implemented optimise on one criteria. The problem in hand (as described above) has three criteria. To deal with this, each algorithm is run three times optimising on a different variable over each separate run. After the route finding algorithms have terminated a set C_{ab} exists containing paths and path description vectors for paths between nodes a and b. Each element in the path description vector is the cumulative value of the corresponding criteria over the entire path from node a to node b. The Pareto Optimal approach is then applied (as described in section 3.1) to construct P_{ab} , the Pareto Optimal Set of journey specifications between the nodes a and b. The path description vectors of candidate solutions are compared with each other rather than with some predefined global optimum path description vector. This is a consequence of the fact that it is not intuitive to define a global optimum vector for paths between any two nodes.

However, this approach has worked very well for static network structures. Static network structures are networks that do not change over time or any changes that occur are separated by a long period of time. This type of static model is mathematically sound but hardly a realistic model of a public transportation network. Transportation networks are inherently dynamic. Changes in route patterns, traffic congestion, road/street availability can change dramatically in a short space of time. Such dynamic changes often render previous estimates of shortest paths or optimal journey specifications incorrect. After a dynamic change, a public transportation information system must quickly update information on shortest/optimal path specification, connectivity structures etc in order to provide up-to-date information to queries. To capture this dynamic behaviour we adopt approaches from the field of dynamic graph algorithms.

4. Dynamic Graph Algorithms

Graph theory and graph data structures and algorithms are inextricably linked with models of any type of physical network. It is well known that computing shortest paths and connectivity relations over a network is the most important task in many network and transportation analyses. Transportation networks possess different levels of congestion, road availability and throughput during different periods of the day. Therefore it is unrealistic to precompute all shortest paths and connectivity at the start of the day and use these to answer queries over the remainder of the day. Shortest paths and other network properties must be updated in real-time that is as soon as a dynamic change occurs on the transportation network. Figure 1 below details the process of updating after a dynamic change. Queries regarding shortest paths or connectivity relations at time $t = T$ are answered regarding the network model in its most up-to-date state.

Dynamic graph data structures and associated algorithms (Eppstein 1998) provide a robust model with which to effectively model the dynamic nature of a public transport network. When a dynamic change fundamentally changes some property or characteristic of the public transport network it is inefficient to re-compute this and related properties from scratch each time. Dynamic information updating (edge congestion, node availability) has a tremendous effect on the planned optimised route causing it, in most cases, to deteriorate. Dynamic algo-

rithms have been shown to be remarkably better than static shortest path algorithms for solving dynamic shortest path problems (Frigioni and Nanni. 1998).

		Dynamic Change	Dynamic Change	QUERY	Dynamic Change
TIME	t = 0	t = t1	t = t2	t = t3	t = t4
ACTION	Build $G = (V,E,S)$	UPDATE	UPDATE		UPDATE
ACTION	Compute Optimal Paths	UPDATE	UPDATE	Return Results	UPDATE

Figure 1: A timeline table of process of dynamic update to a graph model $G = (V,E,S)$ when queries are mixed with update requirements.

5. Graph Mutation

Evolutionary computation described by (Holland 1975) takes a set of candidate solutions to a problem and, using techniques borrowed from natural selection and evolution, evolves these solutions towards ‘fitter’ states. In our case ‘fitter’ states means better solutions in terms of journey specifications. This process has been shown to be efficient at gaining near-optimal solutions to hard problems in polynomial time. The fitness of a set of solutions is evaluated by implementing some decision making process to quantify the relative merit of this evolved set of solutions over the old set of solutions. The evolved set of solutions are deemed ‘fitter’ if and only if they are in some way quantifiably better than the old set of solutions.

This aspect of our evolutionary approach involves making random edge-insertions and edge deletions to the original graph G , producing G' . In the context of a public transportation network this mimics the addition of a route to some previously unused street or road or alternatively the removal of a route link between two nodes. This can be viewed as a random form of route service renewal. All choices regarding edge removal or addition are made randomly. In essence we initiate an unbiased scheme to make alterations to the current graph model of the transportation network. After each change is made a new graph is created. This new graph (a solution) must be evaluated for its fitness to be evolved further. This fitness evaluation requires the Pareto Optimal set of optimal journey specifications to be recomputed. The Pareto Optimal set of the evolved graph and that of the original graph are compared. Based on this comparison conclusions may be drawn on the relative efficiency of the new graph model over the original.

Formally, the insertion or deletion of edges causes the current graph to be mutated into some different graph G' . The resultant candidate set of this graph, C' can be used to produce another Pareto-optimal set P' . If it is the case that the cardinality of the set P' is less than that of P (for a given journey between two nodes) then we know that the random insertion or deletion has produced a better solution than was previously identified. The following example illustrates this (2-d vectors are used for simplicity – modal changes have been omitted):

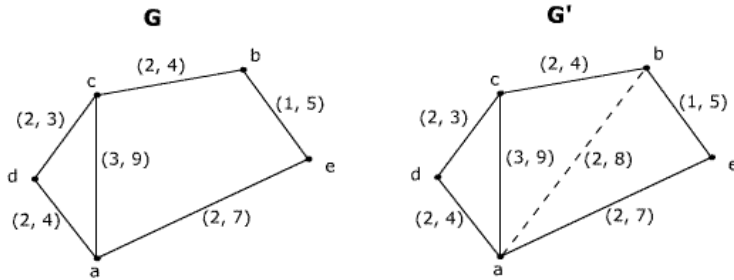


Figure 2: Depicting an original graph model $G = (V,E)$ (with two optimisation criteria). G' is a graph with an edge inserted.

For journeys between node a and b, the following candidate set is produced from the graph G:

Path	Time	Cost
a->d->c->b	6	11
a->e->b	3	12
a->c->b	5	13

Extracting the Pareto-optimal set gives:

Path	Time	Cost
a->d->c->b	6	11
a->e->b	3	12

If a random edge-insertion were to produce G' , the new candidate set produced is:

Path	Time	Cost
a->d->c->b	6	11
a->e->b	3	12
a->c->b	5	13
a->b	2	8

The Pareto-optimal set now becomes:

Path	Time	Cost
a->b	2	8

Members of the Pareto-optimal set for the original graph G are no longer present; it is still possible to use these routes however as the decision maker may choose an alternative route based on other criteria that are not covered by this approach.

The application domain of the graph mutation scheme has, at present, only been applied to public transportation networks in an effort to evolve more efficient network and route structures. Public transportation networks may be viewed as dynamic masks that overlay an existing network (i.e. a road and street network). To this end, this mask may be evolved and changed in an effort to improve the configuration of the network and route structure of the mask without any alteration to the underlying physical network. After a series of evolutions the public transportation network mask (by inspecting the Pareto Optimal Set for journey specifications between nodes) should admit 'better' or more 'fitter' solutions (in terms of journeys and multi-criteria optimisation) than any of its predecessors. This mutant graph (and mask) may then be strongly considered as a replacement for the existing mask. More efficient and better-planned network structures allow the public transportation system operating on this network to operate more effectively. For example there may be a better distribution of routes or a more efficient distribution of these routes causing certain routes to avoid areas of high congestion for example. Also, if the structure of the network is improved route-finding algorithms may find it easier to compute optimal path specifications. The approach of graph mutation is restricted to providing a more efficient and effective structure for journey planning i.e. in regard to network design. It is not necessarily a constituent part of multi-objective analysis.

6. Conclusions and Further Work

The methodology described is designed for implementation as part of the design for Internet based public transportation information systems. Such systems provide infrequent visitors and users of public transportation with a set of optimal journey specifications to choose from during the journey planning process. This set of optimal journey specifications provides valuable guidance during the process of navigating oneself in an unfamiliar urban or suburban environment on a given transportation system. The multi-objective approach to decision making is not in any way confined to the domain of transportation analysis and optimisation. In fact any problem requiring the optimisation of solutions to problems defined in terms of a number of (conflicting or independent) criteria when it is neither possible nor sensible to combine all criteria in some form of aggregation of the criteria.

In this paper we have provided an evolutionary computation framework for the solution of the problem of multi-objective optimisation on a transportation network. This framework has been implemented for optimisation problems involving three criteria. However, more criteria (provided that they can be quantified in some way) may be added without any major changes to the theory. The Pareto Optimal processing stage is $O(n^2)$ in computational complexity and remains so despite the addition of further criteria to the problem. This is a result of the definition of vector domination found in section 3. Preliminary results also found that no particular route finding algorithm was dominant in finding solutions that turned out to be members of the Pareto Optimal set of solutions. Dijkstra's algorithm for example will always find the shortest path (on one cost metric) over a graph structure (Cormen 1999). However there are two reasons why it does not hold a majority on the number of solutions it provides to the Pareto Optimal set. Firstly, while the algorithm will optimise on one criteria or metric the other criteria in the problem are cumulatively gathered and specified in the path description vector. It is then the vector itself (and the tradeoffs between criteria) that determine the solutions suitability for inclusion into the Pareto Optimal set. Secondly, Dijkstra's algorithm, as

well as the other algorithms, performs well on easily quantifiable objectives such as path length, overall path time. However, for criteria that are difficult to quantify i.e. level of convenience of a route or number of modal changes the algorithms perform poorly. This is due to the decision making involved in optimising such quantities. To try to optimise the total number of modal changes one needs to perform some form of look ahead in an attempt to predict possible interchange and connection points further ‘downstream’ of the current node.

The issue of optimisation on a transportation network provides a fertile ground for further research. Our proposal is novel in that it investigates ways of identifying inefficient network structures and deals with multi-objective problems on a dynamic public transportation network. On real-world networks such as these, many of the objectives may be loosely formulated. Examples include, convenience of a route specification, favouritism towards particular routes, road types etc. There has been little research work documented on a crossover approach to optimal design of transportation networks.

References

Coello, C. A. C., “An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends”, Proceedings of the Congress on Evolutionary Computation, Vol. 1, 3-13, IEEE Press, 6-9 July 1999.

Cormen T.H. , *Introduction to Algorithms*. MIT Press, Massachusetts, USA. 1999

Costelloe D, Mooney P and Winstanley A., Multi-Objective Optimisation and Dynamic Routing Algorithms on Public Transportation Networks. *Proc. of GIScience 2001*, Savannah, Georgia USA.

Eppstein D., Dynamic Graph Algorithms. in *Algorithms and Theory of Computation Handbook*, CRC Press, Florida. USA. 1998.

Frigioni D, Nanni U. *Experimental Analysis of Dynamic Algorithms for the Single Source Shortest Path Problem*. Technical Report, University of Rome, “La Sapienza”, Italy. 1998.

Glover, F., Tabu Search: A Tutorial, *Interfaces*, Vol. 20 (1), 74-94, July-August 1990.

Holland J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI 1975.

Metropolis, N and Rosenbluth K., Equations of State Calculations by Fast Computing Machines, *J. Chem. Phys.*, 21, 1087- 1092, 1958.

Nijkamp P and Van Deft A., *Multi-Criteria and Regional Decision Making*, Martinus Nijhoff Science Division, Leiden 1971.